INFORMATION SYSTEM DESIGN IN LARGE SCALE LOGISTIC SYSTEMS

E. P. Durbin

September 1970

P-4308-1

15

INFORMATION SYSTEM DESIGN IN LARGE SCALE LOGISTIC SYSTEMS

E. P. Durbin*
The Rand Corporation
Santa Monica, California

I. INTRODUCTION AND SUMMARY

Operations researchers and systems analysts are increasingly concerned with information system design. Operations research has traditionally been concerned with physical production, distribution, and stockage problems. In these areas operations researchers using techniques such as simulation, Delphi methods, and operational gaming, generally aim at finding strategies -- decisions that take account of important variables at the time the decision is made. The Operations Research approach has not been noticeably successful in improving the Information Systems Design process or the resulting product.

This paper develops the following points:

1. Various factors cause transition to new information systems.
2. The traditional systems analysis approach is a "top-down" design -- specifying goals, objectives, decision variables, policies, and finally information system specifications.
3. Organizations typically follow a "bottom up" or "inside out" design approach.
4. "Bottom up" design occurs because of institutional incentives and because of the complexity of modern systems.
5. The "bottom up" design process typically yields degraded performance.
6. We must consider incremental or phased development that preserves design options.

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The Rand Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The Rand Corporation as a courtesy to members of its staff.

II. LARGE INFORMATION SYSTEMS

An information system is an evolving organization of people, computers, and other equipment (associated communication and support systems) and their integrated operation to regulate and control selected environmental events to achieve systems objectives.

In a complex organization, an information system performs the same function as the nervous system in the human body. This paper deals with information systems used by managers and planners in very large organizations. Such systems may be as simple as item stock level reports in a chain of warehouses or as complex as systems that come into play when an expensive spare part is required by an out-of-commission aircraft at a remote airfield. Stockage, airlift, and procurement information as well as repair computations may be required to determine the point of origin for resupply of the required part. A typical logistics information system consists of several complexes of computers tied together with owned or leased communication facilities. Logistics managers may interact with the information system in making daily decisions, and may enter their decisions back into the information system.

Information system concepts have developed more slowly than hardware. Most attention in large systems has been directed at the transition between second and third generation computer equipment. Second generation systems are characterized by serial memory (tape units), sequential batch processing, and only one user at a time on the central processor. Third generation systems are characterized by direct access memory, various terminal options, multi-programming, and multi-user time sharing.

The complexity of an information system depends on many factors. A few are listed below.

- Nature of Use: An information system used for interactive analysis is more complex than one simply used for data processing and report generation.

- Number of Installations: Complexity increases as the number of interconnected installations increases.

- **Diversity of Equipment:** Problems of standardization and conversion are compounded with diverse equipment.
- **Number of Simultaneous Users:** The executive routines to handle many simultaneous users are complex and costly.
- **Size of Data Base:** Data management systems for very large files and very large records are still being developed.
- **Frequency of Data Changes:** The efficiency of file management systems is increasingly important as the frequency of data updates increases.
- **Interaction of Transactions (Cascading):** Performance of systems in which one event triggers several others is less predictable.
- **Extent of Imbedded Decisionmaking:** Complexity increases as algorithms for decisions are imbedded in the information system and triggered automatically.

III. SYSTEM DESIGN OR CONVERSION

Several factors may lead to initiation of system change -- either design anew or conversion. Workloads increase over time. Facilities begin to wear out and require increased maintenance. Fashions in computing change. Increased operating flexibility is desired. Speed of hardware (not software) improves and arguments concerning decreased dollar costs of each computing operation are difficult to ignore.

Transition may involve shifting transactions from one system to another, perhaps only through software or hardware changes, but perhaps also by policy changes, and perhaps by changing from batch to on-line processing.

Even "simple" hardware or software conversion can involve several considerations. Standardization can be a significant problem in multi-facility system. In converting hardware in such systems we find that physical separation leads to operating differences. This is compounded if generations and manufacturers of equipment are different. File conversion and program conversion are more difficult in a multi-facility system. Staff training in a new system is always required, and planning for additional physical space and power is too often overlooked.

IV. TOP-DOWN DESIGN

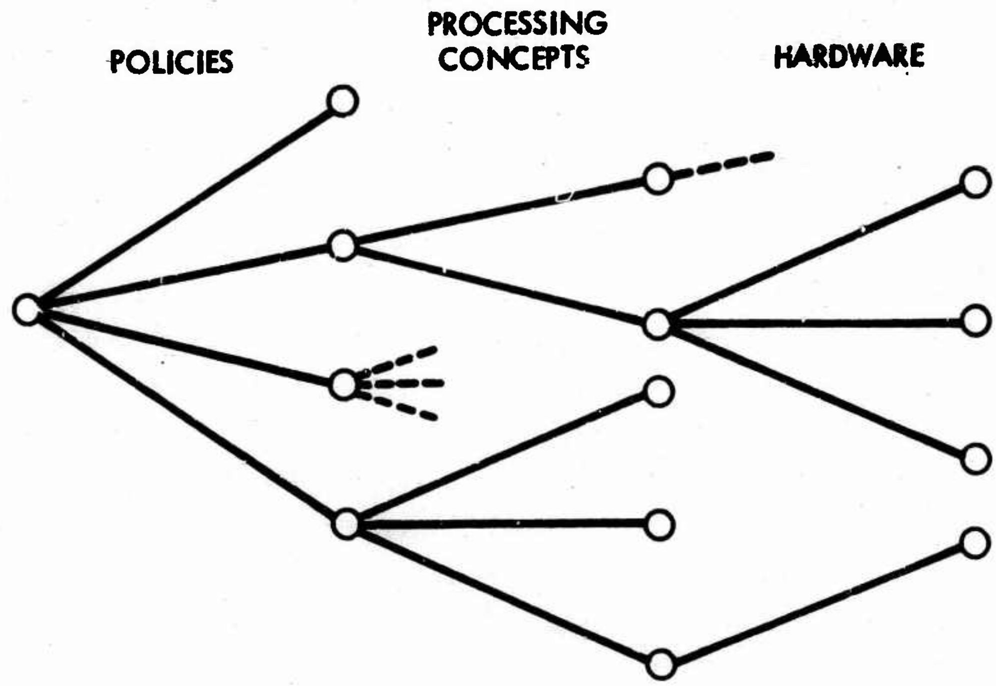
Systems analysts tend to recommend top-down design. Top-down design proceeds from the ideas of constrained optimization. It attempts to look at an overall organization, its policies, and their interactions.

Changes in decision and operating procedures appear to be the source of the major dollar and effectiveness gains in organizations. New technology may be required to implement desired decision and operating procedures, and introduction of new technology may be an essential step. However, resources available for system development are generally limited, often severely. When an initial decision is to take a very large step in introducing new technology, policy improvement will inevitably suffer. The problems associated with simply making new technology run absorb most of the staff. Top-down planning stresses policy and upgrades technology only as necessary. Once a policy base exists, the full range of new technology can be introduced. Processing requirements generate costs. The comparison of policy benefits with processing costs dictates the choice of both policy and information processing schemes. Processing parameters and available technology then lead to hardware selection.

Top-down design typically relies on analytic decision procedures. Such planning emphasizes decision procedures to avoid trouble rather than ad hoc procedures to get out of trouble. Analytic models, simulation, and cost-effectiveness analyses are used to evaluate the worth of policy improvements.

Expending resources on modeling and experimentation requires a tradeoff between time and uncertainty. The more effort put into experimentation and analysis, the greater the reduction in uncertainty about the performance of the ultimate system and reduction in the consequent risk that it will be inadequate. The less effort put into experimentation and analysis, the faster a system gets designed and implemented; but with more attendant uncertainty and risk about ultimate performance. Obviously, when time is available, simulation can be of great benefit.

Top-down design can be likened to a complex decision tree with successive branches in policy, information processing concepts, and hardware configuration.



Policies might include decision procedures for Stock Distribution, Industrial Repair Scheduling, Procurement Policy, or EOQ Purchases. Processing concepts consider the degree of man-machine interaction, mix of batch and on-line processing, types of data base, file management systems, and the degree of system autonomy or manual override. Hardware considerations include the tremendous range of processors, terminals, and storage units available.

Very little is known about where to stop analyzing and experimenting and start implementing. While sound analysis and experimentation is necessary, we can only base our decisions on subjective estimates of the utility of additional research.

Rand's general approach has been to:

(1) Develop formal simulation models of the entire system under consideration to use in evaluating alternative management policies;

(2) Develop detailed simulation or analytical models of the proposed software (data management and multiprogramming systems for example) and experiment with these. For example see how systems behave against different input distributions and study the tradeoffs between data redundancy and information retrieval times.

(3) Develop both gross and detailed simulation models for use in cost-effectiveness and decision-rule studies. These models are elaborations of the system models first developed. They incorporate more of the functional details developed during the system requirements determination and system integration phases and the computer processing details developed during the software requirements development phase. Thus, they are able to attach costs to specific procedures and processing methods and evaluate the benefits achieved through their use.

(4) Survey similar industrial and military systems and collect statistics on software performance. Determine what overhead factors are being incurred and how existing multiprogramming monitors work.

(5) Finally select software structure.

In summary, top-down design of information systems is characterized by a strong degree of sequential decisionmaking based on improved information. It is illustrated in Figure 1 and summarized below.

- Establish organizational missions and goals.
- Develop evaluations of policies through analysis and simulation.
- Explore alternative information processing concepts:
    - Degree of on-line vs. batch processing
    - Degree of man-machine interaction
    - Frequency of update
    - Degree of data base redundancy
- Test a variety of information system options for feasibility
- Compare costs, performances of information system options
- Develop and follow an implementation plan:
    - Delivery, checkout tests
    - Plans for tuning the new system to attain high performance
    - Support, training plans

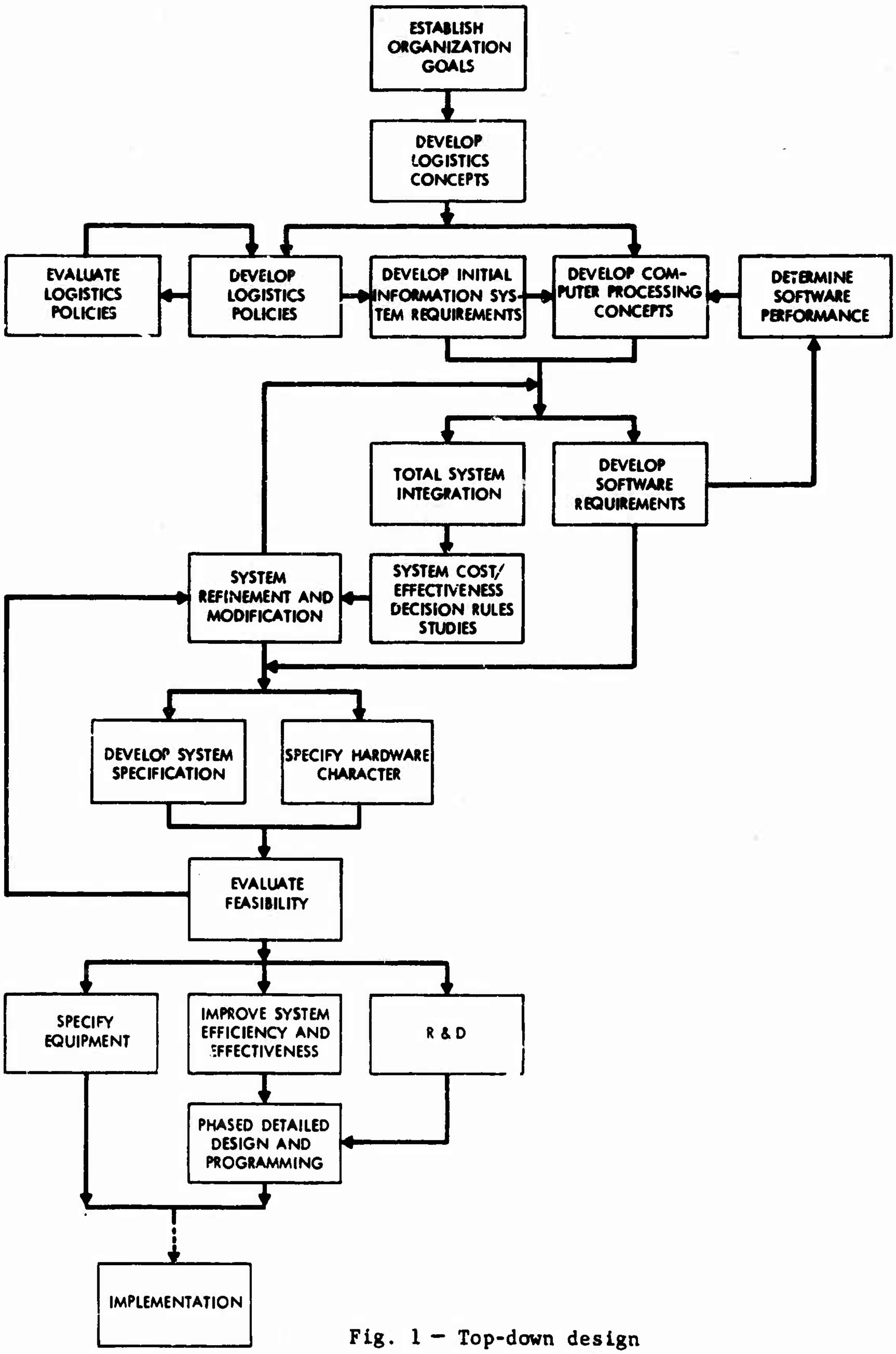


Fig. 1 — Top-down design

V. BOTTOM-UP DESIGN

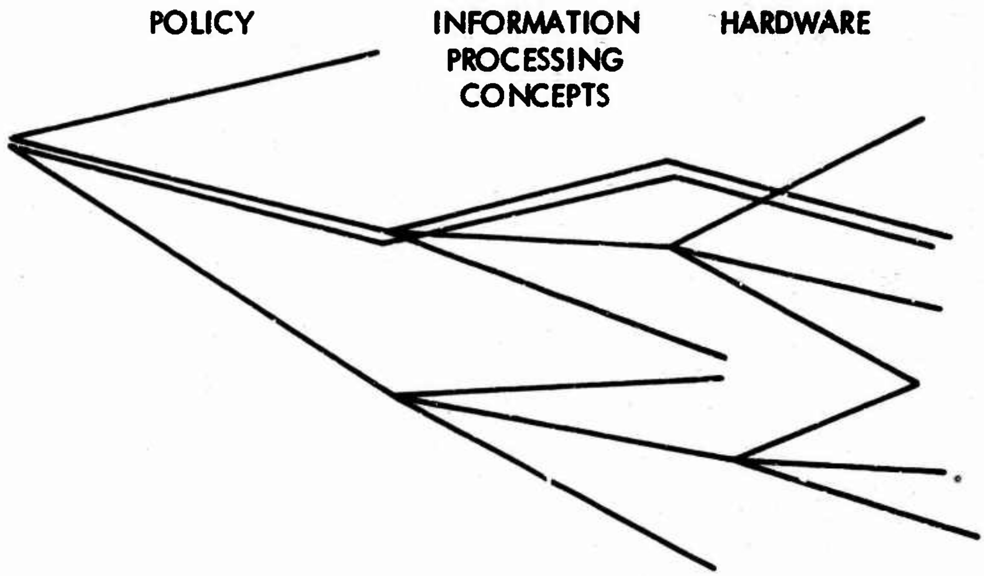
The opposite of the top-down approach is "bottom-up" or "inside-out" analysis. It starts with arbitrary decisions at some detailed level of design, or decisions about specific management policies of the system. It may contain a detailed model of one part of the system, and by a process of addition, build to an overall system picture. As an example, equipment modernization in most corporations has led from 407 punch card equipment to 704 computers to 7090 computers to 360/65 computers with no change in processing rules or frequency or extent of interaction between transactions.

Bottom-up design forces low-level decisions in restricted contexts. It is further characterized by arbitrary selection of hardware. Organization policies are set before any overall planning or cost/effectiveness studies are undertaken, and important policy decisions are made without evaluation of their consequences. Bottom-up decisions frequently reflect a desire to utilize new and perhaps attractive computer hardware rather than a commitment to improvement in the organization's overall performance. Policy innovations receive marginal attention, and enthusiasm is directed at modernizing the processing equipment. Given this initial conceptual set, system design effort tends toward feasibility rather than system performance or cost.

Bottom-up design is frequently also characterized by strong parallel or concurrent structure. Simultaneous choice of management policy and hardware configuration occurs, and software must bridge a possibly unbridgeable gap.

Most design efforts observed in practice appear to initially emphasize estimate system hardware requirements, and later emphasis is on modification to achieve feasibility rather than on design exploration and experimentation to improve organizational performance.

If we characterize the effects of bottom up design on the policy, information concepts, and hardware tree, the tree gets bare rapidly.



VI. DANGER OF BOTTOM-UP DESIGN

Bottom-up design leads to system deficiencies. Arbitrary schedules, policies, development paths, and system boundaries lead to independent inclusion of different policies at different time points without knowledge of their interactions and implications. This can stunt the creativity of designers since there is no overview from which to predict interactions or effects. Thus there is no formal way to introduce new policies, and no valid means of predicting or evaluating policy and system performance.

The rapid and frequently simultaneous pruning of the decision tree in policy and hardware configurations leaves software to fill the gap. This may lead to:

(1) Infeasibility: A management evaluation system may provide no data. Or there may be no interfaces between transactions. Or a communication system may break down under heavy volume.

(2) Performance degradation and system rigidity: After implementation the entire information system cannot be altered. Therefore fewer applications may be run, or less frequent updating than originally planned may be permitted.

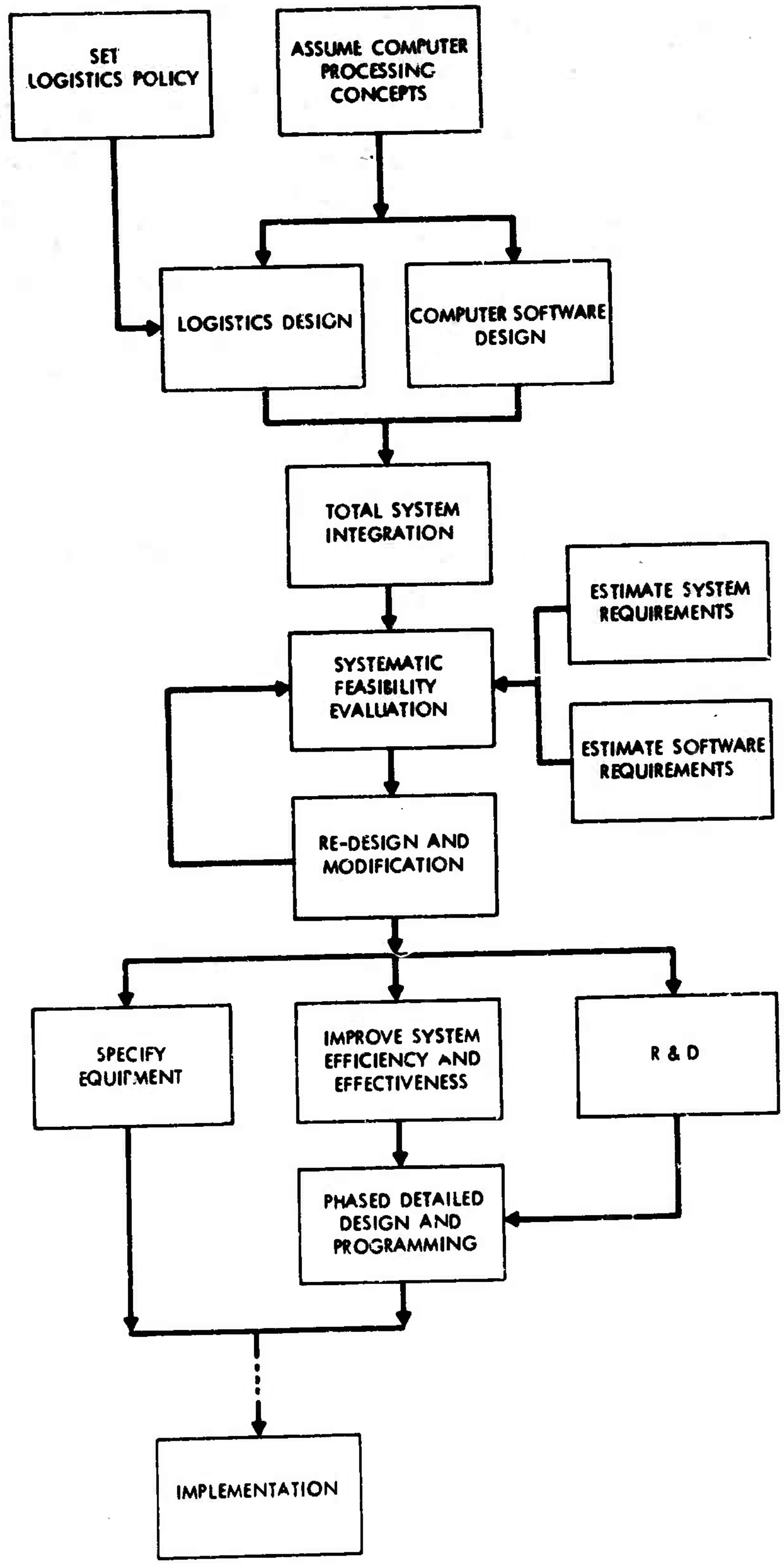


Fig. 2 — Bottom-up design philosophy

(3) Schedule slippage may occur in constructing software or hardware specifications, or in equipment deliveries, or in development of feasible policy applications.

(4) Increased equipment cost may be incurred when extra equipment is procured to permit minimally acceptable system performance.

VII. WHY DOES BOTTOM-UP RATHER THAN TOP-DOWN DESIGN OCCUR?

Bottom-up design is simpler. Top-down design requires determination of organizational and policy goals. These are difficult to obtain. Policy effects and interactions are difficult to model. Moreover, bottom-up design rapidly eliminates uncertainty and yields -- on paper -- a straightforward implementation plan which is easy to monitor. The arbitrariness is unnoticed until too late.

Management is generally not involved in system design. Technicians are typically in actual charge, and it cannot be assumed that the organization's data processing professionals understand corporate managers' responsibilities. It is easier for the design organization to prepare equipment specifications than to assess management needs of diverse organizations.

Procurement mechanisms in large organizations are tedious. System specifications are sometimes desired rapidly. "Buying in" ahead of other capital expenditures in the organization may be desirable. Speed drives the designer to concurrency in policy and hardware selection which requires development of general purpose software independent of any machines. Such software may itself be difficult to develop and probably can contain only a small subset of standard languages.

Bottom-up design flourishes because costs and performance evaluation of system specifications is difficult. The performance of a computer is not determined by either the hardware or the software alone. The performance of an installation (hardware, software, and procedures) depends strongly and markedly on the hardware configuration. Computing standards do not yet exist for many areas. Metrics have not been identified or established. Costing, especially as it relates to procurement, does not reflect true consumption of resources. Because of all

these factors the technical evaluation process is sometimes weak, and lags behind the complexity of systems. Thus, top-down design is not usually used. It is expensive in dollars, time consuming, and may lead to loss of momentum. Short time schedules for system implementation frequently preclude appropriate planning efforts.

VIII. ACHIEVING FEASIBILITY IN THE BOTTOM UP WORLD

In the presence of such realities how may we assure feasibility for large systems?

(1) "Buy it and try it." A common approach, but not good for very large systems since the expense is infeasible.

(2) Tune the existing installation and add equipment as necessary. This is especially attractive since cost-benefit arguments for new systems are generally not borne out.

(3) In designing a new system use experts -- they are much less expensive than in-house personnel. Also modify organizational procedures to fit canned routines and systems (accounting, payroll).

(4) Use a phased, staged, prototype approach to implementation of large systems.

(5) Preserve flexibility and back up by keeping the old system as long as you can, and buying program-compatible equipment.

(6) Rent equipment with an option to return.

(7) If you must buy equipment, buy modular equipment with excess core capacity.

(8) Weigh procurements in favor of vendors with families of equipment.

(9) Following installation, conduct performance audits, tune configurations for additional performance, and return unneeded equipment.

IX. SUMMARY

Top-down design will not in general be accomplished unless a relatively long time for development exists, and an excellent consultant group is on hand. In cases where the developing organization has had information system experience, the design team will probably use an incremental approach rather than a top-down design approach.

Since modern information systems are complex beyond intuition, consultants must realize that simple historical examples and homilies presented to the acquiring organization will not work. Criticism will be ignored. Specific implementable suggestions and specific citations and demonstrations of infeasibility are required.

Consultants must emphasize design for flexibility and change, as well as continuing to advocate modeling and analysis. Rand's empirical studies and observations of past development projects lead us to believe that a highly phased development strategy is preferred. Rather than allocate available resources across many subsystems, focusing resources on one or several critical subsystems has several advantages. First, subsystems are available in the shortest possible time. This strategy permits a staff of modest size and thus a higher quality level can be maintained. Subsystems that appear later in the effort can profit from learning, further policy development, field tests, and simulation exercises. Management and control of phased development are easier since managers do not have to make decisions and follow progress in as many concurrent efforts. Phasing also allows management to recognize that areas differ in terms of (a) payoff, (b) amount of prior work, and (c) ease of development. Phasing does present some difficulties. Some parts of the system must be redesigned and reprogrammed but evidence suggests that the total cost of the phased approach is lower. The real danger of the phased approach is cancellation or loss of momentum prior to completion. Resolution of this problem depends on the organization's procurement policy and on the role taken by top management.

Backup and flexibility must be pressed as key factors. Development is difficult and uncertain. Since management systems always take longer, cost more, and work less well than planned, backup is crucial. Existing systems should be maintained so that they can operate longer if necessary.

Buying new equipment that is program compatible with existing equipment provides extensive backup but may be an unavailable or undesirable option for other reasons. Adequate backup gives the development manager important flexibility. If he encounters a need for modification or additional testing that will delay his program, he can make his decision on the costs and gains involved rather than being forced to meet the schedule. Modularity in design can be achieved by selecting equipment to allow rental or purchase add-ons, to change terminal equipment, and number of peripherals, and to change data transmission volumes. Rental flexibility is especially important since they permit return of parts of the system on short notice.

Prototyping portions of the installation should be encouraged. "System" utilization is a misleading phrase if it is not known which part is critical -- memory, communications, or the CPU. One can install, and measure the utilization rate, with actual loads and then add equipment where necessary. Moreover "system" performance in the abstract generally ignores software and staff skills which are observable in the installation.

Management planning is required to produce the system plan and to build the system. Organization is not a final answer to any problem. but it is important that (1) a strong management role be present throughout development to maintain the policymaking or management function, (2) the project be reviewed at top management level to achieve a cross-function view, and (3) the project group include both functional and computer personnel to allow the close interaction needed in modern systems.

Modern systems analysis is an effort to apply structured rationality to problems of choice. To be of use in Information System Design in large organizations the analyst must be aware that his techniques of analysis require time and data. Neither may be available. The analyst must also understand that institutional factors cause real design to proceed from simultaneous policy and hardware selection through software to the final system. The analyst must therefore supply advice on policy phasing, equipment phasing, flexibility, and backup. Our emphasis must be on creativity-preserving options, and we must plan for the freedom of the next planner of the system.